

APPLICATION
FOR
UNITED STATES LETTERS PATENT

APPLICANT NAME MARK D. SMITH

TITLE SYSTEM AND METHOD FOR DYNAMICALLY UPDATING A SITE MAP
AND TABLE OF CONTENTS FOR SITE CONTENT CHANGES

DOCKET NO. EN999071

INTERNATIONAL BUSINESS MACHINES CORPORATION

CERTIFICATE OF MAILING UNDER 37 CFR 1.10

I hereby certify that, on the date shown below, this correspondence is being deposited with the United States Postal Service in an envelope addressed to the Assistant Commissioner for Patents, Washington, D.C., 20231 as "Express Mail Post Office to Addressee" on 27 Dec 1999

Mailing Label No. EL172581308US

Name of person mailing paper: Judith Anne Beckstrand

Judith Anne Beckstrand
Signature

27 Dec 1999

Date

SYSTEM AND METHOD FOR DYNAMICALLY UPDATING A SITE MAP AND
TABLE OF CONTENTS FOR SITE CONTENT CHANGES

Background of the Invention

Technical Field of the Invention

5 This invention pertains to web technology. More particularly, it relates to the preparation and display of web site maps and tables of contents dynamically updated to current content.

Background Art

10 In a very dynamic web site, such as a Lotus Domino web site, content is added and deleted almost daily, many times hourly. Domino views, or the like, for this content may be categorized by certain topical areas. Content within these topics are added or subtracted by a Content Editing Team.

15 In such an environment, it may be desirable or required to keep a site map and table of contents updated to match the content. Heretofore, it has been required that the site map and table of contents be updated manually. Most, if not all, site maps that exist on web sites are manually updated, 20 or are at such a high level that when content changes, the

site map is still accurate since it does not show low level details.

It is an object of the invention to provide an improved site map and table of contents for a web site.

5 It is a further object of the invention to provide a system and method for providing dynamic updating of a web page site map and table of contents.

10 It is a further object of the invention to provide a system and method enabling a dynamically updated web page site map and table of contents to link directly with volatile content.

It is a further object of the invention to provide a system and a method enabling browser linking of site content to the site map and table of contents.

Summary of the Invention

Sub A web site is broken up into main topical content areas. Each area shows a list of content items that the user can link to, displayed using views. A site map and table of contents read these views to determine what the user should see. The site map is a high level category oriented view, and the table of contents is a more detailed view, getting lower level to content items. Each time the user requests the site map or table of contents, an agent is executed to "lookup" into the content views, giving the user the very latest content.

In accordance with an aspect of the invention, there is provided a computer program product configured to be operable responsive to a request to view a site map and/or table of contents to look into the underlying content views to provide the latest content to the site map and/or table of contents.

Other features and advantages of this invention will become apparent from the following detailed description of the presently preferred embodiment of the invention, taken in conjunction with the accompanying drawings.

Brief Description of the Drawings

Figure 1 is a high level system diagram of the system of the preferred embodiment of the invention for displaying a site view, such as a site map and/or table of contents.

5 Figure 2 is a flow diagram of the method of the preferred embodiment of the invention.

Figure 3 is a screen capture representation of a topical content area page.

10 Figure 4 is a screen capture representation of a site map page.

Figure 5 is a screen capture representation of a table of contents page.

Figure 6 is a screen capture representation of a site map form.

15 Figure 7 is a screen capture representation of a table of contents form.

Best Mode for Carrying Out the Invention

Referring to Figure 1, in accordance with the preferred embodiment of the invention, a site view, such as site map 102 or a table of contents 104, for a web site, such as a Lotus Domino web site, is linked directly to content 106 within the web site using a browser 114.

In a Lotus Domino web site 101 which is very dynamic, content is added to and deleted from a content database 106 almost daily, many times hourly. Domino views 109 for this content are categorized by certain topical areas. Content within these topics are added or subtracted by a Content Editing Team 112. In order to keep a site view or form, such as a Site Map 102 and Table of Contents 104, updated to match the content, and allow a user to link to those content items using a browser 114, a Domino agent 108 is provided to "lookup" into the views 109 that describe the main content areas 106 for new or changed content, and present the most current results to the user 114 in display of site map 116 or TOC 104.

Views 109 are sorted or categorized lists of documents, and are the entry points to data stored in a database 106.

Every database 106 must have at least one view 109, though most have a plurality of views. A database is a container

5 for data, logic, and design elements for an application.

Design elements are building blocks used to create an application, and include pages, forms, outlines, navigators, views, folders, framesets, shared resources, and agents.

Consequently, the agent 108 implementing the preferred

10 embodiment of the present invention is a design. A site map 102 is an outline view.

Forms, like pages, display information. Forms also collect information. A form provides the structure for creating and displaying documents, and documents are the

15 design element that store data in the database.

A field is the part of an application that collects data. Fields can be created on forms, subforms, or in layout regions. Each field stores a single type of information, and a field's field type defines the kind of

20 information a field accepts, such as text, numbers, dates, or names. When a user, either in a client or a browser, creates a form, fills out the information in the fields, and saves the form, the data in the fields is stored in an

individual document. The contents of the fields can then be displayed in documents and views or can be retrieved for use in formulas. When creating a field, the user defines the field name, field type, display options and field properties, computed or editable attribute, and formulas or scripts associated with the field. Items are field data.

Table 1 illustrates the format of a topical content area page.

TABLE 1: TOPICAL CONTENT AREA PAGE

1	Area Category 1	
2	Area 1 content item 1	
3	Area 1 content item 2	
4	. . .	
5	Area 1 content item n	
6	Area Category 2	Descriptive Page Text
7	Area 2 content item 1	
8	Area 2 content item 2	
9	. . .	
10	Area 2 content item n	
11	. . .	
12	Area Category n	
13	Area n content item 1	
14	Area n content item 2	
15	. . .	
16	Area n content item n	

Referring to Figure 3, a sample topical content area page is illustrated.

Table 2 illustrates the format of a site map 102.

TABLE 2: SITE MAP

1	Content Area 1	Content Area 2	...	Content Area N
2	Area 1 Category 1	Area 2 Category 1		Area n Category 1
3	Area 1 Category 2	Area 2 Category 2		Area n Category 2
4
5	Area 1 Category n	Area 2 Category n		Area n Category n

Referring to Figure 4, a sample site map page 102 is illustrated.

Table 3 illustrates the format of a table of contents 104.

TABLE 3: TABLE OF CONTENTS

1	Content Area 1
2	Area 1 Category 1
3	Area 1 content item 1
4	Area 1 content item 2
5	. . .
6	Area 1 content item n
7	. . .
8	Area 1 Category n
9	Area n content item 1
10	Area n content item 2
11	. . .
12	Area n content item n
13	Content Area 2
14	Area 2 Category 1
15	Area 2 content item 1
16	Area 2 content item 2
17	. . .
18	Area 2 content item n
19	. . .
20	Area 2 Category n
21	Area n content item 1
22	Area n content item 2
23	. . .
24	Area n content item n
25	. . .
26	Content Area N
27	Area N Category 1
28	Area N content item 1
29	Area N content item 2
30	. . .
31	Area N content item n
32	. . .
33	Area N Category n
34	Area n content item 1
35	Area n content item 2
36	. . .
37	Area n content item n
38	

Referring to Figure 5, a sample table of contents 104 is illustrated.

When ever an area category or content item is added to content database 106, or deleted, the lists of Tables 1-3 are immediately reflected to the user 114 whenever requested. For example, say in the "Competition" content area, "XYZ Consulting Group" is added. The next time the user looks at the Table of Contents 104, "XYZ Consulting Group" would appear there in "Competition" - no "design change" or manual update to the table of contents 104 is needed.

The site map 102 and TOC forms 104 are very similar. Referring to Figure 6, a site map form 102 is illustrated. Referring to Figure 7, a table of contents form 104 is illustrated.

Referring to Figure 6, there are five basic elements to the structure of the site map form 102, as follows:

1. Layout structure, in tabular form, for the site map.
2. Title "Site Map".
3. Header and footer, pulled in from the web site.

4. Form type identifier, SiteMap (used by the CreateMap agent of Table 4, infra).
5. Data field 116, filled in by CreateMap agent 108 with site map data dynamically when the page 116 is requested by the user.

Referring to Figure 7, there are five basic elements to the structure of the table of contents (TOC) form 104, as follows:

1. Layout structure, in a single column, for the TOC.
- 10 2. Title "Table of Contents"
3. Header and footer, pulled in from the web site.
4. Form type identifier, TOC (used by the CreateMap agent of Table 4, infra).
5. Data field 118, filled in by CreateMap agent 108 with TOC data dynamically when the page 104 is requested by the user.

Referring to Figure 2, in accordance with a preferred embodiment of the method of the invention, in step 120 the agent 108 to create the appropriate data for the site map 102 or table of contents (TOC) 104 first initializes variables that will be needed within the agent 108. In step 122, the agent 108 sets up the start of layout 113, 117 for

either the site map or table of contents, depending on the form being request. If it is a site Map, an HTML table structure 115 is set up, and if it is a TOC, an HTML list structure 118 is set up. In step 124, the agent gets access 5 to the navigation view 110 for the site. In step 126, for each navigation document 111 in the view, the agent 108 determines the appropriate category name and, in step 128, the agent adds an HTML list item 115 or 118 for the category and a URL to link within the site 101. Continuing in step 10 128, if this execution of the agent 108 is with respect to the TOC 104, a list item 118 is added for the document name and the URL to a location on site 101. In step 130, the agent closes table cells as appropriate, and in step 132 handles any special case(s) that may need to be added. In 15 step 134, all of this data 115, 118 is moved and saved to a field 116, 118 on the appropriate form 102, 104 for display.

Table 4 sets forth the agent code for the create map procedure of the preferred embodiment of the invention, as described above with respect to Figure 2.

TABLE 4 : CREATE MAP AGENT

```
1 Sub Initialize
2     Dim session As New notessession
3     Dim db As notesdatabase
4     Dim view As notesview
5     Dim doc As notesdocument
6     Dim doc2 As notesdocument
7     Dim dc As notesdocumentcollection
8     Dim mainview As NotesView
9
10    Set db = session.currentdatabase
11    Set doc2 = session.documentcontext
12
13    If doc2.MapType(0) = "sitemap" Then
14        tablestart = "[<TABLE CELLPADDING=1 CELLSPACING=0
15        BORDER=1>]"
16        tableend = "[</TABLE>]"
17        cellstart = "[<TD ALIGN=LEFT VALIGN=TOP><img
18        src=""//cons/home/spacer.gif"" width=130 height=1
19        border=0>]"
20        cellend = "[</TD>]"
21    Else
22        tablestart = ""
23        tableend = ""
24        cellstart = ""
25        cellend = ""
26    End If
27
28    Set mainview = db.GetView("layerMap")
29    Set doc = mainview.getfirstdocument
30    category = ""
31    sub_category = ""
32    maincatnum = 0
33
34    While Not doc Is Nothing
35        If category <> doc.Category_Name(0) Then
36            category = doc.Category_Name(0)
37            Select Case Lcase(category)
38            Case "refs"
39                viewtitle = "Our Customers"
40                viewurl =
41                    "/home.nsf/references?OpenView"
42            Case "e-biz"
43                viewtitle = "e-business Essentials"
```

```

44         viewurl = "/essentials"
45     Case "presentations"
46         viewtitle = "Presentations"
47         viewurl =
48             "/home.nsf/presentations?OpenView"
49     Case "news"
50         viewtitle = "News"
51         viewurl = "/news"
52     Case "geographies"
53         viewtitle = "Geographies"
54         viewurl = "/home.nsf/ebusgeo?openview"
55     Case "comp"
56         viewtitle = "Competition"
57         viewurl =
58             "/home.nsf/competition?OpenView"
59     Case "salesres"
60         viewtitle = "More Resources"
61         viewurl =
62             "/home.nsf/Sales+Resources?OpenView"
63     Case "ebkb"
64         viewtitle = "e-business Value
65             Knowledgebase"
66         viewurl = "/knowledgebase.nsf"
67 End Select
68 If maincatnum = 0 Then
69     DisplayField = DisplayField + tablestart
70 Else
71     DisplayField = DisplayField + cellend
72 End If
73 maincatnum = maincatnum + 1
74 DisplayField = DisplayField + cellstart +
75 "[</ul><p></font><font size=4><a href=" +
76 viewurl + ">" + viewtitle + "</a></font><font
77 size=-1><ul>]"
78 sub_category = doc.SubCategory(0)
79 catnum = 0
80 If sub_category <> "" Then
81     If doc2.MapType(0) = "sitemap" Then
82         catnum = catnum + 1
83         url = viewurl + "&Expand=" +
84             Cstr(catnum) + "#" + Cstr(catnum)
85         DisplayField = DisplayField +
86             {[</ul><li><a href="} + url + {">]}
87             + sub_category + {[</a><ul>]}
88 Else
89     DisplayField = DisplayField +
90         "[</ul><li><b>]" + sub_category +
91         "[</b><ul>]"
92     DisplayField = DisplayField +

```

```

93                     "[<li>]" + doc.Title(0)
94             End If
95         Else
96             DisplayField = DisplayField + "[<li>]" +
97                 doc.Title(0)
98         End If
99     Else
100        If sub_category <> doc.SubCategory(0) Then
101            sub_category = doc.SubCategory(0)
102            If doc2.MapType(0) = "sitemap" Then
103                catnum = catnum + 1
104                url = viewurl + "&Expand=" +
105                    Cstr(catnum) + "#" + Cstr(catnum)
106                DisplayField = DisplayField +
107                    {[</ul><li><a href="" + url + ">"]}
108                    + sub_category + {[</a><ul>]}
109            Else
110                DisplayField = DisplayField +
111                    "[</ul><li><b>]" + sub_category +
112                    "[</b><ul>]"
113                DisplayField = DisplayField +
114                    "[<li>]" + doc.Title(0)
115            End If
116        Else
117            If sub_category = "" Then
118                DisplayField = DisplayField +
119                    "[<li>]" + doc.Title(0)
120            Else
121                If doc2.MapType(0) = "toc" Then
122                    DisplayField = DisplayField +
123                        "[<li>]" + doc.Title(0)
124                End If
125            End If
126        End If
127    End If
128
129    Set doc = mainview.getnextdocument (doc)
130 Wend
131
132    'Handle Priority Offerings (Solutions) SPECIAL
133
134    'Priority Offerings (Solutions)
135    viewurl = "/home.nsf/globale-bussol?ReadForm"
136    viewtitle = "Solutions"
137    solutions = "[</ul><p></font><font size=4><a href=" +
138        viewurl + ">" + viewtitle + "</a></font><font size=-
139        1><ul>]"
140    DisplayField = DisplayField + cellend + cellstart +
141    solutions

```

```
142 'e-commerce
143 viewurl = "/ecomm.nsf/ECOMhome?ReadForm"
144 viewtitle = "E-commerce"
145 ecomm = {[<li><a href="} + viewurl + {">]} + viewtitle
146 + {[</a>]}
147 DisplayField = DisplayField + ecomm
148 'ERP
149 viewurl = "/erp"
150 viewtitle = "ERP"
151 erp = {[<li><a href="} + viewurl + {">]} + viewtitle +
152 {[</a>]}
153 DisplayField = DisplayField + erp
154
155 DisplayField = DisplayField + "[</font></ul></ul>]" +
156 cellend + tableend
157 Call doc2.ReplaceItemValue("DisplayMap",DisplayField)
158
159 End Sub
```

Advantages over the Prior Art

It is an advantage of the invention that there is provided an improved site map and table of contents for a web page.

5 It is an advantage of the invention that there is provided a system and method for providing dynamic updating of a web page site map and table of contents.

It is an advantage of the invention that there is

provided a system and method enabling a dynamically updated web page site map and table of contents to link directly with volatile content.

It is an advantage of the invention that there is
5 provided a system and a method enabling browser linking of site content to the site map and table of contents.

Alternative Embodiments

It will be appreciated that, although specific embodiments of the invention have been described herein for
10 purposes of illustration, various modifications may be made without departing from the spirit and scope of the invention. In particular, it is within the scope of the invention to provide a computer program product or program element, or a program storage or memory device such as a
15 solid or fluid transmission medium, magnetic or optical wire, tape or disc, or the like, for storing signals readable by a machine, for controlling the operation of a computer according to the method of the invention and/or to structure its components in accordance with the system of
20 the invention.

Further, each step of the method may be executed on any general computer, such as an IBM System 390, AS/400, PC or the like and pursuant to one or more, or a part of one or 5 more, program elements, modules or objects generated from any programming language, such as C++, Java, Pl/I, Fortran or the like. And still further, each said step, or a file or object or the like implementing each said step, may be executed by special purpose hardware or a circuit module 10 designed for that purpose.

Accordingly, the scope of protection of this invention is limited only by the following claims and their equivalents.